

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN VIỄN THÔNG
-----o0o-----



ĐỒ ÁN CƠ SỞ NGÀNH

HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ QUA INTERNET

GVHD: Võ Tuấn Kiệt

SVTH: Tạ Duy Khiêm

MSSV: 2211575

TP. HỒ CHÍ MINH, THÁNG 3 NĂM 2025

LỜI CẢM ƠN

Trước tiên, em xin chân thành cảm ơn thầy Võ Tuấn Kiệt đã hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đồ án này. Thầy không chỉ truyền đạt kiến thức chuyên sâu mà còn luôn động viên, khích lệ em vượt qua những khó khăn trong việc nghiên cứu và thực hiện dự án.

Nhờ có sự hướng dẫn tận tình của thầy, em đã có được cái nhìn sâu sắc về lĩnh vực cảm biến và vi điều khiển, cũng như những kỹ năng cần thiết để phát triển một hệ thống đo nhịp tim hiệu quả. Sự chia sẻ kinh nghiệm và những phản hồi quý báu từ thầy đã giúp em hoàn thiện đồ án một cách tốt nhất.

Em cũng xin cảm ơn các bạn đồng nghiệp và những người đã hỗ trợ em trong quá trình thực hiện đồ án. Hy vọng rằng những kiến thức và kinh nghiệm tích lũy được từ đề tài này sẽ giúp em trong các nghiên cứu và dự án tiếp theo.

Một lần nữa, em xin chân thành cảm ơn thầy!

Tp. Hồ Chí Minh, ngày 18 tháng 03 năm 2025 .

Sinh viên

Tạ Duy Khiêm

TÓM TẮT ĐỒ ÁN

Đồ án này trình bày về việc phát triển một hệ thống giám sát nhiệt độ và điều khiển thiết bị sử dụng cảm biến DHT11 và vi điều khiển ESP8266, với khả năng điều khiển quạt và còi báo khi nhiệt độ vượt ngưỡng cho phép. Hệ thống được thiết kế nhằm cung cấp một giải pháp đơn giản và hiệu quả trong việc theo dõi và kiểm soát nhiệt độ tự động, giúp nâng cao sự an toàn và tiện lợi trong môi trường sử dụng.

Đề tài bắt đầu với việc nghiên cứu nguyên lý hoạt động của cảm biến DHT11, cách thức giao tiếp với ESP8266, cũng như phương pháp điều khiển quạt và còi báo động. Tiếp theo, mạch điện được thiết kế và lập trình phần mềm được thực hiện để thu thập dữ liệu nhiệt độ, xử lý và kích hoạt thiết bị phù hợp. Qua nhiều giai đoạn thử nghiệm, hệ thống đã cho thấy tính ổn định và hiệu quả trong việc điều chỉnh nhiệt độ môi trường.

Kết quả của đồ án không chỉ góp phần tối ưu hóa việc kiểm soát nhiệt độ trong các ứng dụng thực tiễn như nhà thông minh hay hệ thống làm mát tự động, mà còn mở ra hướng đi cho các ứng dụng IoT trong giám sát và điều khiển từ xa. Đồ án này hy vọng sẽ là nền tảng cho các nghiên cứu và ứng dụng tiếp theo trong việc phát triển các hệ thống tự động thông minh.

Mục Lục

1. Giới Thiệu	4
1.1. Tổng quan	4
1.2. Nhiệm vụ đề tài.....	4
2. Lý Thuyết.....	5
2.1. Giới thiệu về Blynk IoT	5
2.1.1. Quy trình sử dụng Blynk IoT:	6
2.2. Giới thiệu về kit ESP8266.....	7
2.2.1. Cấu tạo chính của kit ESP8266:	8
2.2.2. Nguyên lý hoạt động của kit ESP 8266:.....	11
2.3. Giới thiệu cảm biến DHT11	12
2.3.1. Cấu tạo của cảm biến DHT11:	13
2.3.2. Nguyên lý hoạt động của cảm biến DHT11:.....	14
3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG.....	15
3.1. Yêu cầu thiết kế	15
3.1.1. Các yêu cầu đặt ra	15
3.2. Phân tích thiết kế.....	16
3.2.1. Phương pháp thiết kế dựa trên yêu cầu đã đặt ra	16
3.2.2 So sánh và lựa chọn phương án thiết kế	18
3.3 Sơ đồ khối:	19
Khối điều khiển quạt	20
3.4. Tính toán và vẽ sơ đồ mạch:.....	21
3.4.1. Schematic:.....	21
3.4.2. Tính toán các khối trong mạch:	21
3.5. Lưu đồ giải thuật:.....	25
4. Kết quả thực hiện:	26

4.1. Thi công phần cứng.....	26
4.2 Thi công phần mềm:	28
4.3. Các bước thực hiện:	34
4.4 Kết quả thực nghiệm:	39
5. Kết luận:	41
5.1 Kết luận.....	41
5.2 Hướng phát triển.....	42
5.3 Ứng dụng thực tiễn :	43
6. Tài liệu tham khảo:.....	44

1. Giới Thiệu

1.1. Tổng quan

Trong những năm gần đây, công nghệ IoT (Internet of Things) đã được ứng dụng rộng rãi trong giám sát và điều khiển tự động. Đặc biệt, trong lĩnh vực môi trường và thiết bị điện, việc giám sát nhiệt độ và điều khiển tự động các thiết bị như quạt và cảnh báo bằng còi buzzer giúp tối ưu hóa hiệu suất và tiết kiệm năng lượng.

Hệ thống trong đề tài này sử dụng vi điều khiển ESP8266 NodeMCU để thu thập dữ liệu từ cảm biến nhiệt độ và độ ẩm DHT11. Dữ liệu thu thập được sẽ hiển thị trên nền tảng Blynk IoT, cho phép người dùng giám sát từ xa thông qua internet. Khi nhiệt độ vượt ngưỡng cài đặt, hệ thống sẽ tự động kích hoạt quạt làm mát và cảnh báo bằng buzzer để người dùng nhận biết và có thể điều chỉnh thiết bị một cách hợp lý.

1.2. Nhiệm vụ đề tài

Nội dung 1: Tìm hiểu nguyên lý và lý thuyết về đo nhiệt độ, độ ẩm

Mục tiêu đầu tiên là nghiên cứu nguyên lý hoạt động của cảm biến DHT11, bao gồm cách đo nhiệt độ và độ ẩm, cũng như các yếu tố ảnh hưởng đến độ chính xác của cảm biến. Sinh viên cần thu thập tài liệu khoa học, nghiên cứu các phương pháp đo và so sánh DHT11 với các cảm biến tương tự khác.

Nội dung 2: Tìm hiểu về vi điều khiển ESP8266 và nền tảng Blynk.iot

Nhiệm vụ tiếp theo là nghiên cứu về vi điều khiển ESP8266 NodeMCU, cách lập trình và giao tiếp với các cảm biến và thiết bị ngoại vi. Đồng thời, sinh viên sẽ tìm hiểu về nền tảng Blynk IoT, cách thiết lập ứng dụng trên điện thoại, gửi và nhận dữ liệu từ xa thông qua giao thức MQTT hoặc HTTP.

Nội dung 3: Thiết kế và lập trình hệ thống điều khiển tự động

Hệ thống sẽ được thiết kế với các thành phần chính bao gồm cảm biến DHT11, quạt làm mát, buzzer cảnh báo và đèn LED báo trạng thái. Vi điều khiển ESP8266 sẽ xử lý dữ liệu và điều khiển thiết bị thông qua chương trình được viết bằng Arduino IDE. Mục tiêu là tạo ra một hệ thống hoạt động ổn định, có khả năng giám sát từ xa và phản hồi nhanh với các điều kiện môi trường thay đổi.

Nội dung 4: Kiểm tra, đánh giá và tối ưu hệ thống

Cuối cùng, hệ thống sẽ được thử nghiệm thực tế để kiểm tra độ chính xác của cảm biến và hiệu quả của việc điều khiển tự động. Dữ liệu được ghi nhận và phân tích để đánh giá hiệu suất của hệ thống so với các thiết bị đo thương mại. Nếu cần thiết, hệ thống sẽ được tối ưu hóa để cải thiện độ chính xác và tốc độ phản hồi.

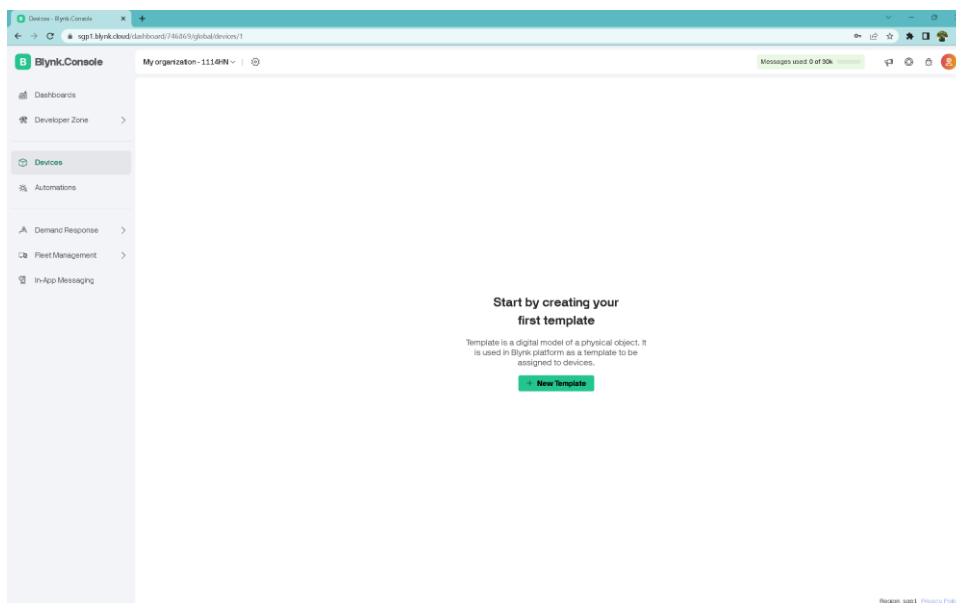
Kết luận

Hệ thống giám sát và điều khiển nhiệt độ sử dụng ESP8266 và Blynk IoT giúp nâng cao hiệu quả trong việc quản lý môi trường. Với khả năng giám sát từ xa, hệ thống có thể ứng dụng trong nhà thông minh, nông nghiệp và các khu vực cần kiểm soát nhiệt độ tự động. Kết quả của đề tài không chỉ giúp tiết kiệm năng lượng mà còn mở ra hướng nghiên cứu ứng dụng IoT trong các lĩnh vực khác.

2. Lý Thuyết

2.1. Giới thiệu về Blynk IoT

Blynk IoT là một nền tảng điện toán đám mây và ứng dụng di động, cho phép người dùng giám sát và điều khiển các thiết bị IoT từ bất kỳ đâu thông qua internet. Nền tảng này cung cấp giao diện trực quan, giúp người dùng dễ dàng theo dõi dữ liệu từ cảm biến, điều chỉnh cài đặt và nhận cảnh báo từ hệ thống một cách nhanh chóng và thuận tiện.



Hình 1: Trang chính của Blynk IoT trên máy tính

2.1.1. Quy trình sử dụng Blynk IoT:

Đăng ký và tạo tài khoản trên Blynk IoT:

Tải ứng dụng Blynk từ cửa hàng trên điện thoại (Android hoặc iOS). Hoặc truy cập trang web tại [Blynk.cloud](https://blynk.cloud)

Đăng ký tài khoản hoặc đăng nhập nếu đã có.



Hình 2: Ứng dụng Blynk trên điện thoại.

Tạo dự án mới:

Mở ứng dụng Blynk, chọn tạo dự án mới.

Đặt tên dự án và chọn loại bo mạch (ví dụ: ESP8266).

Nhận mã xác thực (Auth Token) được gửi qua email; mã này sẽ dùng trong code của ESP8266.

Lập trình cho ESP8266:

Cài đặt thư viện Blynk trong Arduino IDE (hoặc môi trường phát triển khác).

Viết chương trình cho ESP8266 bao gồm:

- Nhập Auth Token, thông tin Wi-Fi (SSID và mật khẩu).
- Thiết lập kết nối Wi-Fi và liên kết với máy chủ Blynk.
- Đọc dữ liệu từ các cảm biến (như DHT11) và gửi dữ liệu lên Blynk, đồng thời nhận lệnh điều khiển từ ứng dụng.

Nạp chương trình vào bo mạch ESP8266.

Thiết kế giao diện trên ứng dụng Blynk:

Sử dụng giao diện kéo thả của Blynk để thêm các widget (ví dụ: gauge, button, LED, chart) theo nhu cầu hiển thị và điều khiển.

Cấu hình các widget để liên kết với các chân ảo (virtual pins) đã định nghĩa trong code.

Kết nối và giám sát:

Khi ESP8266 đã chạy chương trình và kết nối Wi-Fi thành công, dữ liệu từ thiết bị sẽ được gửi đến máy chủ Blynk.

Mở ứng dụng Blynk trên điện thoại để theo dõi trạng thái thiết bị theo thời gian thực và gửi các lệnh điều khiển (như bật/tắt quạt, kích hoạt buzzer, v.v.).

Tối ưu và mở rộng:

Theo dõi và điều chỉnh các widget, lập trình dựa trên phản hồi từ hệ thống để cải thiện hiệu suất giám sát và điều khiển.

Có thể mở rộng tính năng bằng cách thêm các thiết bị hoặc cảm biến mới vào dự án.

2.2. Giới thiệu về kit ESP8266

ESP8266 là một vi điều khiển tích hợp module WiFi có kích thước nhỏ gọn và chi phí thấp, được thiết kế để phục vụ các ứng dụng IoT. Với khả năng kết nối không dây, ESP8266 cho phép giao tiếp và xử lý dữ liệu từ các cảm biến như DHT11, từ đó điều khiển các thiết bị ngoại vi như quạt, buzzer và LED. Việc tích hợp ESP8266 giúp đơn giản hóa quá trình phát triển các hệ thống giám sát và điều khiển tự động từ xa.

2.2.1. Cấu tạo chính của kit ESP8266:

a) Mặt trước:



Hình 3: mặt trước của kit ESP8266

Mặt trước của bo mạch ESP8266 (phiên bản NodeMCU) trong hình có các đặc điểm nổi bật sau:

Module WiFi ESP-12 (ESP8266MOD)

Trung tâm của bo mạch là module WiFi ESP8266 được che bởi vỏ kim loại hình chữ nhật, trên đó in thông tin như logo Wi-Fi, chứng chỉ FCC, dải tần 2.4GHz.

Phần ăng-ten dạng đường mạch in (PCB antenna) nằm ở mép trên của vỏ kim loại, có hình ziczac đặc trưng.

Cổng USB Type-C

Ở phía dưới bo mạch là cổng USB Type-C (thay vì micro USB như nhiều phiên bản cũ), dùng để cấp nguồn và nạp chương trình cho vi điều khiển.

Dãy chân GPIO hai bên

Hai bên bo mạch là các hàng chân GPIO:

Tên chân	Chức năng chính	Chức năng đặc biệt
3V3	Cấp nguồn 3.3V	-
Vin	Cấp nguồn đầu vào (5V)	-
RST	Reset	Khởi động lại mạch
D0 (GPIO16)	GPIO	Wake-up từ chế độ Deep Sleep
D1 (GPIO5)	GPIO, I2C SCL	-
D2 (GPIO4)	GPIO, I2C SDA	-
D3 (GPIO0)	GPIO, Flash mode	Chân chọn chế độ nạp (BOOT)
D4 (GPIO2)	GPIO, UART TX1, LED	Chân LED tích hợp trên bo mạch
D5 (GPIO14)	GPIO, SPI SCK	-
D6 (GPIO12)	GPIO, SPI MISO	-
D7 (GPIO13)	GPIO, SPI MOSI	-
D8 (GPIO15)	GPIO, SPI CS	Chân phải kéo xuống GND khi khởi động
RX (GPIO3)	UART RX	Nhận dữ liệu từ Serial
TX (GPIO1)	UART TX	Truyền dữ liệu từ Serial
A0 (ADC0)	Chân analog	Đọc giá trị điện áp (0 - 1V)
GND	Mass (0V)	-
EN (CH_PD)	Enable	Kéo lên 3.3V để bật ESP8266

Các chân này cho phép người dùng kết nối với cảm biến, module mở rộng và các thiết bị ngoại vi khác.

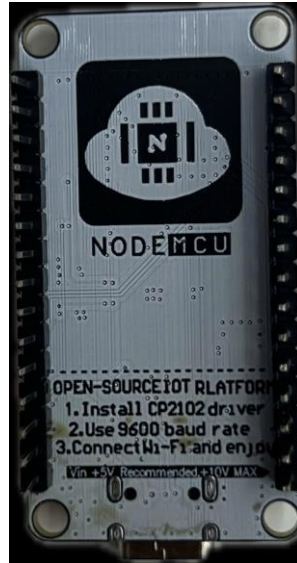
Nút RST và FLASH

Thường được bố trí gần cổng USB hoặc góc bo mạch (có thể thấy nút RST và nút FLASH nhỏ màu đen), dùng để khởi động lại và chuyển mạch sang chế độ nạp firmware.

Linh kiện SMD và mạch cấp nguồn

Trên bo mạch có các linh kiện dán (SMD) như điện trở, tụ điện, IC ổn áp 3.3V, chip chuyển đổi USB-to-Serial (ở trong kit này sử dụng CP2102) giúp giao tiếp với máy tính qua cổng USB. Các linh kiện này được bố trí gọn gàng xung quanh khu vực module ESP8266.

b) Mặt sau:



Hình 4: Mặt sau của kit ESP8266

Mặt sau của bo mạch ESP8266 NodeMCU có thiết kế in logo đặc trưng của NodeMCU, kèm theo một số hướng dẫn sử dụng cơ bản. Dưới đây là các chi tiết trên mặt sau:

Hướng dẫn sử dụng (OPEN-SOURCE IOT PLATFORM):

Install CP2102 driver: Hướng dẫn người dùng cài đặt driver CP2102 (chip chuyển đổi USB-UART) để giao tiếp với máy tính.

Use 9600 baud rate: Đề xuất sử dụng tốc độ truyền dữ liệu 9600 baud trong giao tiếp serial.

Connect Wi-Fi and enjoy: Nhắc nhở về khả năng kết nối Wi-Fi để phát triển ứng dụng IoT.

Thông tin về nguồn cấp điện:

Vin +5V Recommended +10V MAX:

Chân Vin có thể nhận điện áp từ 5V đến tối đa 10V.

Tuy nhiên, khuyến nghị cấp 5V để đảm bảo an toàn và ổn định cho mạch.

Lỗ hàn và đường mạch:

Mặt sau có các đường mạch in giúp kết nối linh kiện trên bo mạch.

Một số điểm hàn được thiết kế sẵn để mở rộng tính năng hoặc sửa chữa nếu cần.

2.2.2. Nguyên lý hoạt động của kit ESP 8266:

1) Cấp nguồn và ổn áp

Khi mạch được cấp nguồn (qua cổng USB hoặc chân VIN), điện áp được ổn định xuống mức 3.3V bởi bộ ổn áp tích hợp. Đây là bước khởi đầu đảm bảo rằng ESP8266 và các linh kiện trên bo mạch nhận được nguồn điện phù hợp.

2) Khởi động và nạp bootloader

Sau khi điện áp ổn định, ESP8266 sẽ khởi động và chạy bootloader nội bộ.

Bootloader kiểm tra trạng thái của các chân GPIO (như GPIO0, GPIO2, GPIO15) để xác định chế độ hoạt động: chạy chương trình hoặc vào chế độ nạp firmware. Nếu không có yêu cầu nạp lại firmware, bootloader sẽ chuyển quyền điều khiển cho chương trình đã được lưu trữ trong bộ nhớ flash.

3) Tải và thực thi chương trình từ bộ nhớ flash

Chương trình (firmware) do người dùng phát triển được lưu trữ trong bộ nhớ flash.

ESP8266 tải chương trình này vào bộ nhớ và bắt đầu thực thi các lệnh theo thứ tự đã định sẵn trong code.

4) Thiết lập kết nối Wi-Fi

Trong quá trình khởi tạo, chương trình cấu hình module Wi-Fi tích hợp của ESP8266.

Nó sẽ quét và kết nối với mạng Wi-Fi theo các thông số (SSID, mật khẩu) đã được cài đặt. Khi kết nối thành công, ESP8266 có thể giao tiếp qua Internet hoặc với các thiết bị khác.

5) Thực hiện các tác vụ ứng dụng

Sau khi kết nối mạng, ESP8266 thực hiện các tác vụ chính của ứng dụng. Điều này có thể bao gồm:

Đọc dữ liệu từ các cảm biến (ví dụ: DHT11 cho nhiệt độ và độ ẩm).

Xử lý dữ liệu và thực hiện các điều khiển như bật quạt, kích hoạt buzzer, hay đèn LED.

Giao tiếp với nền tảng IoT: như Blynk IoT để giám sát từ xa hoặc gửi dữ liệu lên máy chủ.

Tạo web server: hoặc sử dụng các giao thức giao tiếp như MQTT để trao đổi dữ liệu.

Vòng lặp chính và giám sát liên tục: ESP8266 chạy một vòng lặp chính (loop) liên tục để cập nhật và giám sát trạng thái của hệ thống. Trong vòng lặp này, nó thường xuyên kiểm tra dữ liệu từ cảm biến, xử lý logic ứng dụng và gửi/nhận thông tin qua mạng. Nếu phát hiện lỗi hoặc nhận lệnh reset, quá trình khởi động lại sẽ được thực hiện, đưa mạch trở về trạng thái ban đầu.

2.3. Giới thiệu cảm biến DHT11

DHT11 là một cảm biến nhiệt độ và độ ẩm nhỏ gọn, thường được sử dụng để đo lường điều kiện môi trường trong các ứng dụng gia đình và công nghiệp. Cảm biến này hoạt động dựa trên nguyên lý thay đổi điện trở của các thành phần cảm biến khi nhiệt độ và độ ẩm thay đổi, qua đó chuyển đổi dữ liệu thành tín hiệu số để xử lý và hiển thị.

2.3.1. Cấu tạo của cảm biến DHT11:

Cảm biến DHT11 được cấu tạo thành một module tích hợp.



Hình 5: Cảm biến DHT11

Cấu tạo của cảm biến DHT11 bao gồm:

Vỏ bọc bảo vệ:

Một lớp vỏ nhựa hoặc epoxy giúp bảo vệ các thành phần bên trong khỏi tác động của môi trường và các nhiễu ngoại lai.

Phần cảm biến độ ẩm:

Sử dụng một lớp polymer nhạy cảm với độ ẩm, tạo thành một bộ điện dung.

Khi độ ẩm thay đổi, điện dung của lớp polymer cũng thay đổi, cho phép xác định giá trị độ ẩm của không khí.

Phần cảm biến nhiệt độ:

DHT11 sử dụng một thermistor (cảm biến điện trở nhiệt), trong đó điện trở thay đổi theo nhiệt độ môi trường.

Sự thay đổi này được chuyển đổi thành tín hiệu số để xác định giá trị nhiệt độ.

Mạch xử lý tín hiệu:

Một vi mạch nhỏ tích hợp, chịu trách nhiệm chuyển đổi các tín hiệu analog từ cảm biến thành tín hiệu kỹ thuật số.

Mạch này cũng đảm bảo tính chính xác thông qua cơ chế kiểm tra (checksum) khi truyền dữ liệu.

Các chân kết nối: Có 3 chân:

- VCC: Nguồn cấp cho cảm biến (thường 3.3V hoặc 5V).
- DATA: Chân truyền nhận dữ liệu số giữa cảm biến và vi điều khiển.
- NC: Chân không sử dụng (trong một số module có thể không xuất hiện).

2.3.2. Nguyên lý hoạt động của cảm biến DHT11:

Cảm biến DHT11 là một module đo nhiệt độ và độ ẩm kỹ thuật số, hoạt động dựa trên nguyên lý biến đổi điện dung và điện trở của các thành phần cảm biến:

1. Đo độ ẩm:

DHT11 sử dụng một cảm biến độ ẩm điện dung.

Một lớp polymer nhạy cảm với độ ẩm được phủ lên bề mặt cảm biến. Khi không khí có độ ẩm thay đổi, điện dung của lớp polymer này cũng thay đổi theo.

Mạch bên trong DHT11 chuyển đổi sự thay đổi điện dung này thành tín hiệu điện tử.

2. Đo nhiệt độ:

Cảm biến nhiệt độ của DHT11 dựa trên nguyên lý hoạt động của thermistor, tức là cảm biến điện trở nhiệt.

Giá trị điện trở của thermistor thay đổi theo nhiệt độ môi trường.

Sự thay đổi điện trở được chuyển đổi thành tín hiệu kỹ thuật số biểu thị nhiệt độ.

3. Truyền dữ liệu:

Sau khi đo lường, DHT11 sử dụng giao thức truyền thông một dây (single-wire protocol) để gửi dữ liệu nhiệt độ và độ ẩm.

Dữ liệu được truyền đi dưới dạng 40 bit (5 byte): 2 byte dành cho độ ẩm, 2 byte dành cho nhiệt độ và 1 byte kiểm tra tính chính xác (checksum).

Qua đó, DHT11 cung cấp thông tin nhiệt độ và độ ẩm bằng cách chuyển đổi các thay đổi vật lý (điện dung và điện trở) thành tín hiệu số, giúp các vi điều khiển như ESP8266 dễ dàng thu thập và xử lý dữ liệu để ứng dụng trong các dự án IoT.

3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

3.1. Yêu cầu thiết kế

3.1.1. Các yêu cầu đặt ra

Đo lường môi trường:

Nhiệt độ: Đo trong khoảng từ 0°C đến 50°C, với độ chính xác $\pm 2^\circ\text{C}$.

Độ ẩm: Đo trong khoảng từ 20% đến 90% RH, với độ chính xác $\pm 5\%$ RH.

Xử lý và giao tiếp:

Vi điều khiển: Sử dụng ESP8266 NodeMCU, hoạt động ở mức điện áp 3.3V, có tích hợp WiFi (hỗ trợ chuẩn 802.11 b/g/n) và cung cấp ít nhất 8 chân GPIO cho các thiết bị ngoại vi.

Giao tiếp với cảm biến: Kết nối cảm biến DHT11 qua một chân dữ liệu số (1-wire protocol).

Kết nối mạng: Hệ thống phải có khả năng kết nối WiFi ổn định, đảm bảo khoảng cách hoạt động ít nhất 30m trong điều kiện nội thất.

Giám sát và hiển thị:

Nền tảng IoT: Tích hợp với Blynk IoT để giám sát và điều khiển từ xa.

Giao diện người dùng: Ứng dụng Blynk cần hiển thị dữ liệu nhiệt độ, độ ẩm và trạng thái hệ thống, đồng thời gửi cảnh báo khi dữ liệu vượt ngưỡng.

Tần số cập nhật: Dữ liệu được cập nhật liên tục, ví dụ cập nhật mỗi 1 giây.

Cảnh báo:

Ngưỡng nhiệt độ cảnh báo: Khi nhiệt độ vượt quá 36°C.

Ngưỡng độ ẩm cảnh báo: Khi độ ẩm quá 60% RH.

Cơ chế cảnh báo: Gửi thông báo qua Blynk và kích hoạt cảnh báo (ví dụ: buzzer hoặc LED báo trạng thái).

Nguồn cấp và ổn áp:

Nguồn cấp: Cấp nguồn từ cổng USB hoặc nguồn ngoài ổn định, đảm bảo đầu ra là 3.3V cho ESP8266.

Bảo vệ nguồn: Sử dụng bộ ổn áp và các linh kiện bảo vệ (như diode, tụ điện) để đảm bảo hoạt động ổn định và an toàn.

Chi phí và khả năng mở rộng:

Chi phí: Giải pháp cần tối ưu về chi phí, sử dụng linh kiện giá thành thấp nhưng vẫn đảm bảo độ tin cậy.

Khả năng mở rộng: Hệ thống có thể dễ dàng bổ sung các cảm biến hoặc module khác nếu cần trong tương lai.

3.2. Phân tích thiết kế

3.2.1. Phương pháp thiết kế dựa trên yêu cầu đã đặt ra

Lựa chọn cảm biến: DHT11

Cách thức hoạt động: DHT11 sử dụng nguyên lý thay đổi điện dung của lớp polymer nhạy với độ ẩm và sự thay đổi điện trở của thermistor để đo nhiệt độ.

Ưu điểm: Giá thành thấp, dễ sử dụng, module tích hợp sẵn.

Khuyết điểm: Phạm vi đo và độ chính xác hạn chế (chỉ phù hợp với ứng dụng giám sát môi trường thông thường).

Định lượng: Đo nhiệt độ từ 0°C đến 50°C ($\pm 2^\circ\text{C}$); đo độ ẩm từ 20% đến 90% ($\pm 5\%$ RH).

Lựa chọn vi điều khiển: ESP8266 NodeMCU

Cách thức hoạt động: ESP8266 có tích hợp WiFi, cho phép kết nối internet và giao tiếp với cảm biến thông qua các chân GPIO. Chương trình được nạp qua giao tiếp USB-to-Serial và chạy từ bộ nhớ flash.

Ưu điểm: Giá thành rẻ, dễ lập trình (hỗ trợ Arduino IDE), có khả năng kết nối không dây mạnh mẽ.

Khuyết điểm: Số chân GPIO hạn chế so với các vi điều khiển hiện đại như ESP32; bộ nhớ và tốc độ xử lý không cao.

Định lượng: Hoạt động ở 3.3V; WiFi hoạt động theo chuẩn 802.11 b/g/n; cập nhật dữ liệu từ cảm biến mỗi 1 giây.

Lựa chọn nền tảng giám sát từ xa: Blynk IoT

Cách thức hoạt động: Blynk IoT cung cấp giao diện di động để người dùng giám sát dữ liệu từ hệ thống, gửi lệnh điều khiển và nhận cảnh báo thông qua internet.

Ưu điểm: Giao diện thân thiện, dễ cấu hình, cho phép giám sát thời gian thực từ xa; phiên bản miễn phí đủ dùng cho các ứng dụng nhỏ.

Khuyết điểm: Phụ thuộc vào kết nối internet và máy chủ của Blynk; một số hạn chế về số lượng dự án trong phiên bản miễn phí.

Định lượng: Tần số cập nhật dữ liệu khoảng 1 giây; thông báo khi nhiệt độ $> 40^\circ\text{C}$ hoặc độ ẩm $< 30\%$ RH.

3.2.2 So sánh và lựa chọn phương án thiết kế

Lựa chọn kit:

Yếu tố	ESP8266	ESP32 (Sự lựa chọn thay thế)
Khả năng kết nối	WiFi tích hợp	WiFi + Bluetooth
Số chân GPIO	Hạn chế	Nhiều hơn
Giá thành	Rẻ hơn	Cao hơn
Ứng dụng	Ứng dụng đơn giản, giám sát cơ bản	Ứng dụng yêu cầu xử lý phức tạp hơn

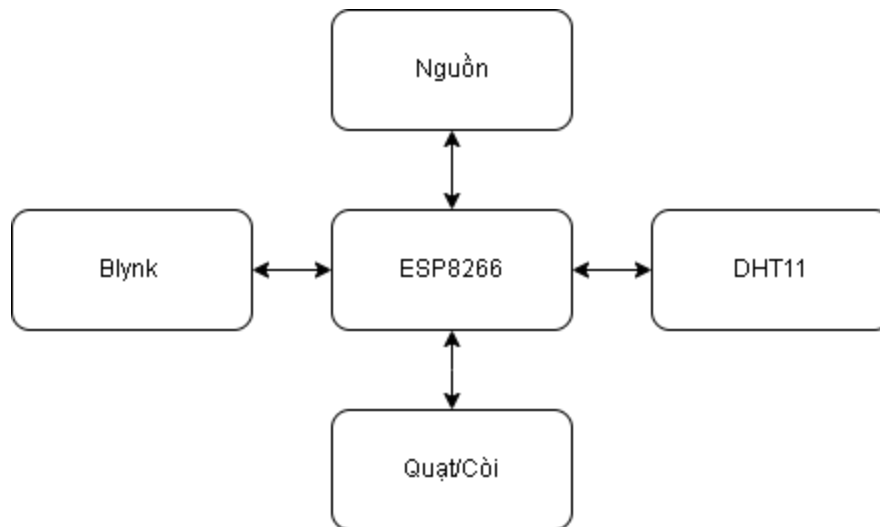
Lựa chọn cảm biến:

Yếu tố	DHT11	DHT22 (Sự lựa chọn thay thế)
Độ chính xác	$\pm 2^{\circ}\text{C}$, $\pm 5\%$ RH	$\pm 0.5^{\circ}\text{C}$, $\pm 2-5\%$ RH (độ chính xác cao hơn)
Phạm vi đo	$0^{\circ}\text{C} - 50^{\circ}\text{C}$, 20% - 90% RH	-40°C đến 80°C , 0% - 100% RH
Giá thành	Rất rẻ	Cao hơn
Ứng dụng	Ứng dụng giám sát cơ bản	Ứng dụng đòi hỏi độ đo chính xác cao

Lựa chọn phần mềm IoT:

Yếu tố	Blynk IoT	Nền tảng thay thế (ThingsBoard,...)
Giao diện	Thân thiện, dễ cấu hình	Có thể phức tạp hơn trong cấu hình
Phụ thuộc Internet	Có (khả năng giám sát từ xa)	Có
Chi phí	Miễn phí với giới hạn nhỏ	Miễn phí bản cơ bản nhưng có hạn chế

3.3 Sơ đồ khối:



Mạch gồm các khối chính:

Khối nguồn cấp

Thành phần:

- Điện áp đầu vào (nguồn ngoài).
- Diode chỉnh lưu 1N4007 bảo vệ ngược cực.
- IC ổn áp LM7805 tạo ra nguồn 5V ổn định.
- Tụ điện 1000 μ F, 104 giúp lọc nhiễu nguồn.
- LED báo nguồn để kiểm tra trạng thái hoạt động.

Chức năng: Cung cấp nguồn 5V ổn định cho vi điều khiển và các linh kiện khác.

Khối vi điều khiển (ESP8266 NodeMCU)

Thành phần:

- ESP8266 NodeMCU có các chân kết nối:
- DHT11 (chân DATA) để nhận dữ liệu nhiệt độ, độ ẩm.
- Quạt (qua transistor Q1) để điều khiển bật/tắt.
- Buzzer (còi báo) để cảnh báo khi nhiệt độ cao.
- LED báo trạng thái để hiển thị hoạt động.

Chức năng: Xử lý tín hiệu từ cảm biến và điều khiển thiết bị theo điều kiện nhiệt độ.

Khối cảm biến DHT11

Thành phần: Cảm biến DHT11 với 3 chân:

VCC: Cấp nguồn 3.3V – 5V.

GND: Nối đất.

DATA: Gửi dữ liệu nhiệt độ/độ ẩm về ESP8266.

Chức năng: Đo nhiệt độ và độ ẩm môi trường, gửi dữ liệu về ESP8266 để xử lý.

Khối điều khiển quạt

Thành phần:

Transistor Q1 đóng ngắt quạt.

Diode 1N4007 bảo vệ dòng ngược.

Điện trở R2 (1K) giúp điều chỉnh dòng.

LED1 hiển thị trạng thái bật/tắt quạt.

Chức năng: Khi nhiệt độ cao, ESP8266 kích transistor để cấp nguồn cho quạt chạy.

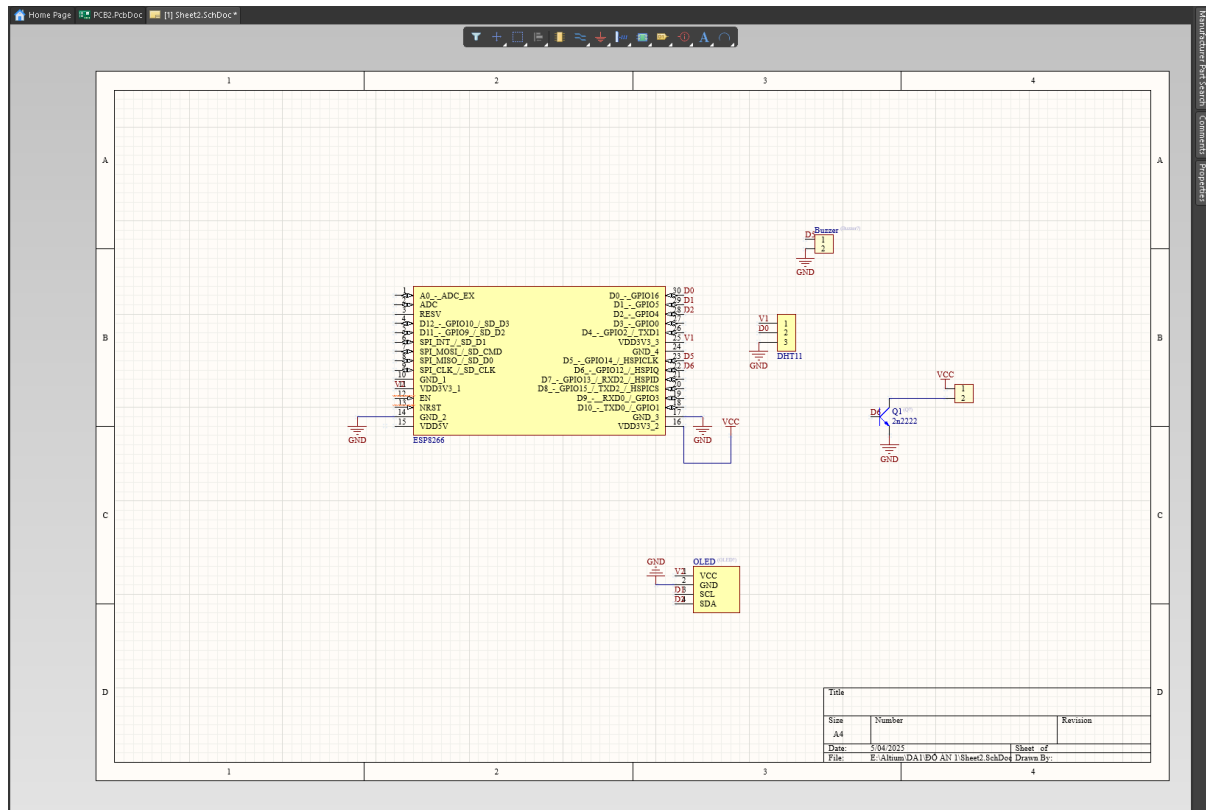
Khối còi báo (Buzzer)

Thành phần: Một còi buzzer đơn giản.

Chức năng: Khi nhiệt độ vượt quá ngưỡng nguy hiểm, ESP8266 sẽ kích hoạt còi để phát cảnh báo.

3.4. Tính toán và vẽ sơ đồ mạch:

3.4.1. Schematic:



Hình 6: Sơ đồ mạch trên Altium

3.4.2. Tính toán các khối trong mạch:

Khối Cảm biến DHT11

Yêu cầu và thông số kỹ thuật:

Phạm vi đo nhiệt độ: 0°C đến 50°C (độ chính xác $\pm 2^{\circ}\text{C}$)

Phạm vi đo độ ẩm: 20% đến 90% RH (độ chính xác $\pm 5\%$ RH)

Nguồn cấp: 3.3V – 5V

Tốc độ cập nhật: Thông thường khoảng 1 giây/lần

Tiêu thụ điện năng: Rất thấp (thường dưới 0.3 mA khi đo)

Tính toán hỗ trợ giao tiếp:

Điện trở kéo lên cho chân DATA:

Để đảm bảo tín hiệu ổn định cho giao tiếp 1 dây, cần có điện trở kéo lên (thường từ 4.7 kΩ đến 10 kΩ) nối từ chân DATA đến VCC.

Chọn 10 kΩ, với điện áp 5V thì dòng kéo qua điện trở là:

$$I = \frac{V}{R} = \frac{5}{10000} = 0.5mA$$

⇒ Dòng này đủ nhỏ để không ảnh hưởng đến hoạt động của cảm biến.

Khối Vi điều khiển ESP8266 NodeMCU

Yêu cầu và thông số kỹ thuật:

Điện áp hoạt động: 3.3V (được cấp từ nguồn 5V thông qua mạch ổn áp nội bộ)

Tiêu thụ điện năng: Trong khoảng 80 – 200 mA khi WiFi hoạt động (giá trị trung bình khoảng 150 mA)

Tốc độ xử lý: CPU Tensilica LX106, đủ khả năng xử lý dữ liệu từ cảm biến và kết nối WiFi

Tính toán về nguồn và tiêu thụ:

Nguồn cấp điện nội bộ:

Nếu ESP8266 được cấp từ nguồn 5V (ví dụ qua cổng USB), thì mạch ổn áp nội bộ sẽ chuyển đổi xuống 3.3V.

Hiệu suất ước tính của một bộ ổn áp tuyến tính (linear regulator) là:

$$H = \frac{3.3}{5} \times 100\% \approx 66\%$$

Công suất tiêu thụ:

Giả sử trung bình tiêu thụ 150 mA ở 3.3V:

$$P = V \times I = 3.3 \times 0.15 = 0.495W$$

Điều này cần được xem xét trong thiết kế giải nhiệt của mạch ổn áp.

Khối Nguồn Cấp và Ổn Áp

Yêu cầu và thông số kỹ thuật:

Nguồn vào: 5V (qua USB hoặc adapter)

Ổn áp: Chuyển đổi 5V xuống 3.3V cho ESP8266

Bảo vệ mạch: Sử dụng diode và tụ điện lọc nhiễu để đảm bảo điện áp ổn định

Tính toán cơ bản:

Điện áp rơi trên bộ ổn áp:

$$V_{roi} = V_i - V_o = 5 - 3.3 = 1.7V$$

Công suất tiêu hao trên bộ ổn áp:

Với dòng tối đa giả định 200 mA:

$$P_{ht} = V_{roi} \times I = 1.7 \times 0.2 = 0.34W$$

Giá trị này cần được đảm bảo không gây quá nhiệt cho bộ ổn áp.

Khối Cảnh Báo (Buzzer) và Điều Khiển Quạt

Ví dụ tính toán cho mạch điều khiển quạt qua transistor:

Yêu cầu: Kích hoạt quạt khi nhiệt độ vượt ngưỡng (giả sử quạt hoạt động ở 5V với dòng 200 mA).

Bộ chuyển mạch: Sử dụng transistor NPN

Tính toán điện trở Base cho transistor:

Giả sử:

- Điện áp xuất ra từ GPIO của ESP8266: 3.3V
- Điện áp nối vào Base-Emitter (V_{BE}): khoảng 0.7V
- Giả sử cần dòng vào base: khoảng 2 mA (để đảm bảo transistor bão hòa với hệ số khuếch đại $h_{FE} \sim 100$)

Tính điện trở Base:

$$R_B = \frac{V_{GPIO} - V_{BE}}{I_B} = \frac{3.3 - 0.7}{0.002} = \frac{2.6}{0.002} = 1300\Omega$$

Khởi Giám sát qua Blynk IoT

Yêu cầu và thông số kỹ thuật:

Tần số cập nhật dữ liệu: 1 giây/lần

Dung lượng dữ liệu truyền: Mỗi lần cập nhật dữ liệu chỉ truyền vài byte (thông tin nhiệt độ, độ ẩm và trạng thái hệ thống)

Băng thông yêu cầu: Rất thấp, không vượt quá vài kilobyte mỗi giờ.

Tính toán băng thông:

Giả sử: Mỗi gói dữ liệu có kích thước khoảng 64 byte.

Cập nhật 1 gói mỗi giây:

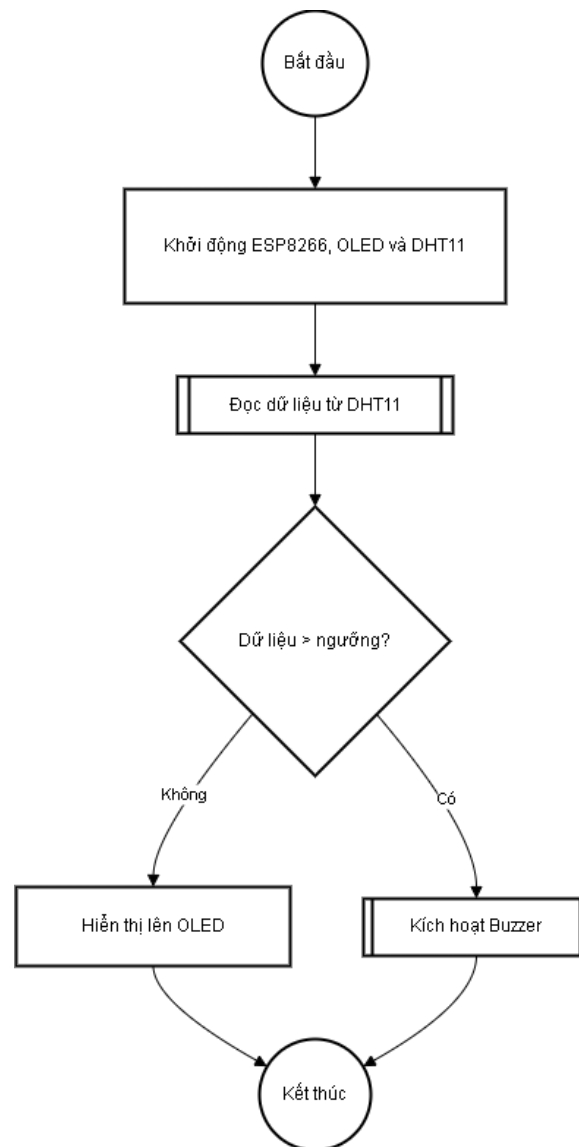
$$64 \times 1 = 64 \text{ (byte/s)}$$

Trong 1 giờ (1h = 3600s):

$$64 \times 3600 = 230400 \text{ byte} \approx 230\text{KB}$$

Băng thông này hoàn toàn nằm trong mức tiêu thụ của các kết nối mạng tiêu chuẩn cho ứng dụng IoT.

3.5. Lưu đồ giải thuật:



Bắt đầu: Khởi động hệ thống.

Khởi động ESP8266, OLED và DHT11:

ESP8266: Vi điều khiển kết nối Internet, đóng vai trò xử lý trung tâm.

OLED: Màn hình hiển thị thông tin (nhiệt độ/độ ẩm).

DHT11: Cảm biến đo nhiệt độ và độ ẩm môi trường.

Đọc dữ liệu từ DHT11: Thu thập giá trị nhiệt độ/độ ẩm từ cảm biến.

Kiểm tra điều kiện "Dữ liệu > ngưỡng?":

So sánh giá trị đọc được với một ngưỡng định trước (ví dụ: nhiệt độ > 30°C).

Nếu đúng (vượt ngưỡng), hệ thống thực hiện:

- Hiển thị lên OLED: Hiển thị thông số vượt ngưỡng trên màn OLED.
- Kích hoạt Buzzer: Bật còi cảnh báo.

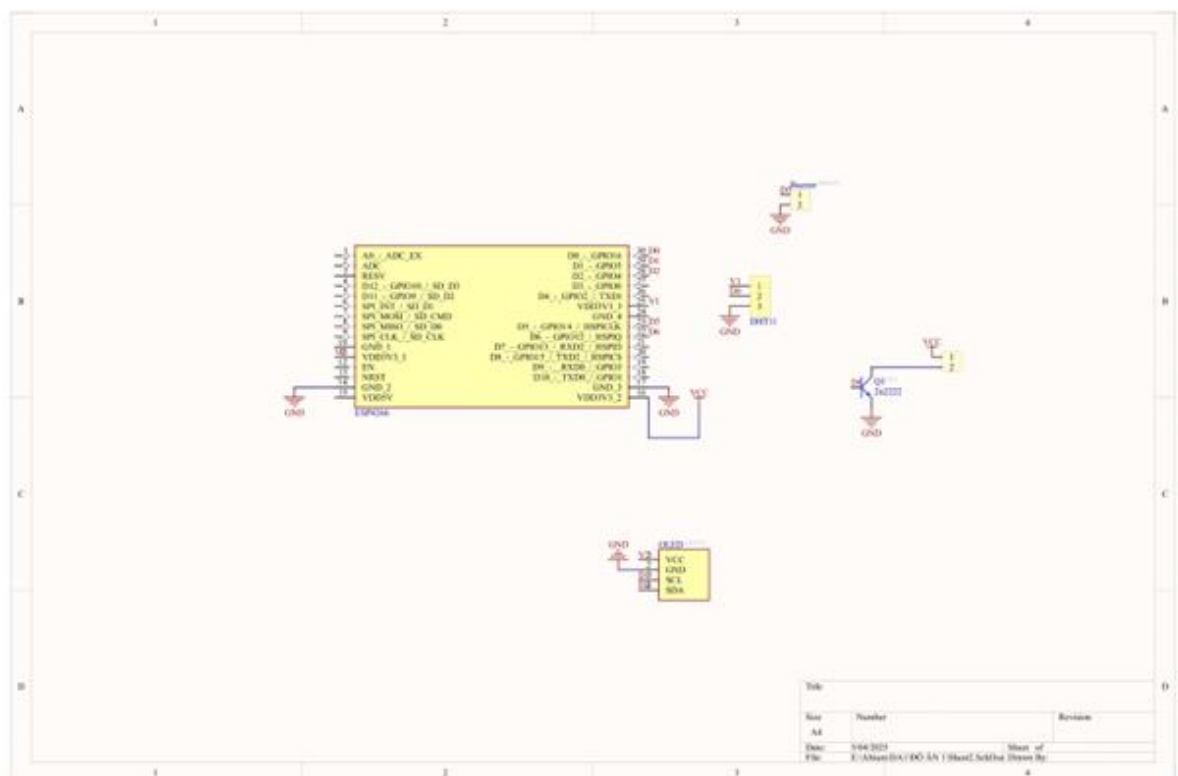
Nếu sai, hệ thống có thể lặp lại quá trình đọc dữ liệu.

Kết thúc: Kết thúc quy trình (hoặc quay lại bước đọc dữ liệu để tiếp tục giám sát).

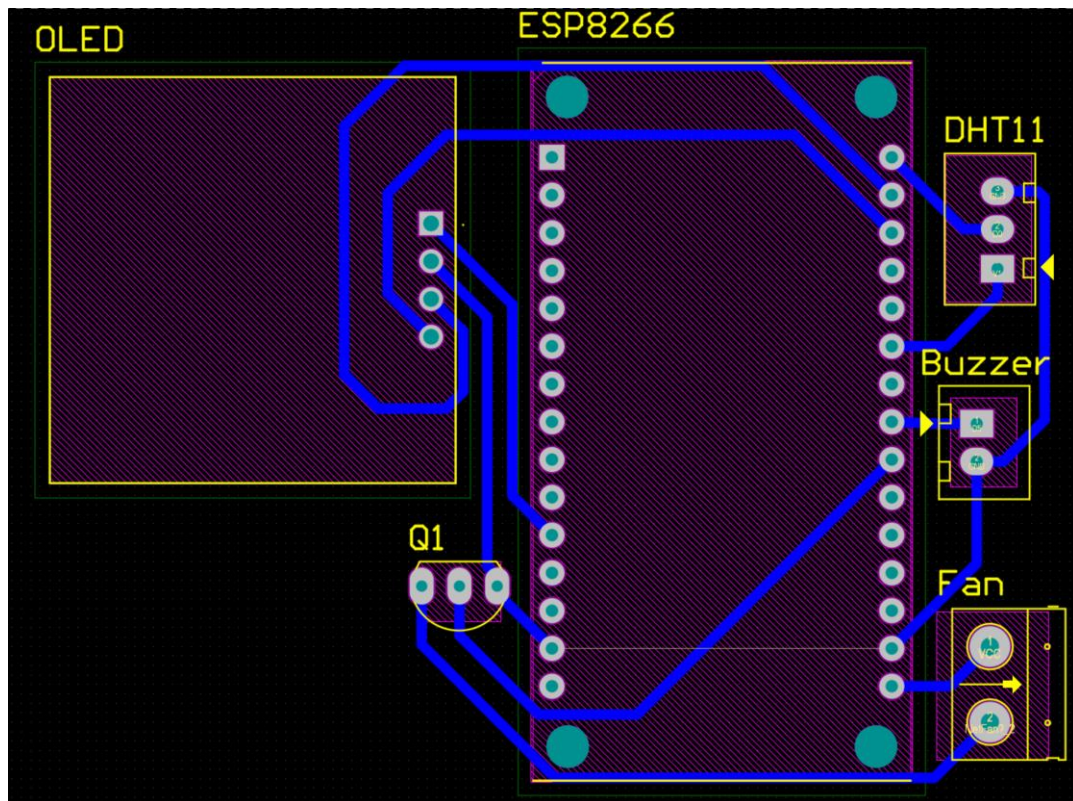
4. Kết quả thực hiện:

4.1. Thi công phần cứng

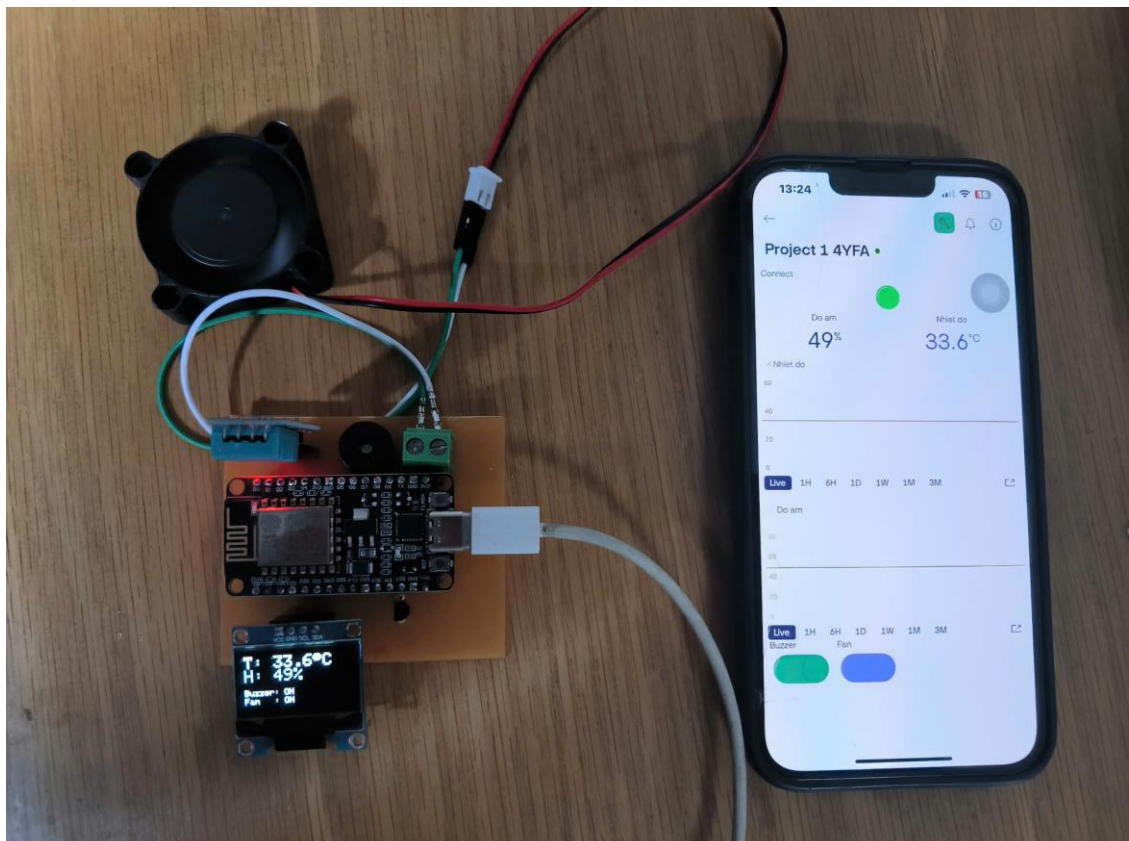
Đã hoàn thành thiết kế sơ đồ nguyên lý (schematic):



Thiết kế Layout của hệ thống:



4. Đã lắp ráp các thành phần phần cứng như kit ESP8266, cảm biến DHT11, quạt, buzzer và màn hình OLED.



4.2 Thi công phần mềm:

Đã hoàn thành nạp code vào mạch:

```

1 #define BLYNK_TEMPLATE_ID "TMPL6yV917LqZ"
2 #define BLYNK_TEMPLATE_NAME "Project 1"
3 #define BLYNK_FIRMWARE_VERSION "0.1.0"
4 #define BLYNK_PRINT Serial
5 #define APP_DEBUG
6 #define USE_NODE_MCU_BOARD
7 #include "BlynkEdgent.h"
8 #include "DHTesp.h"
9 #include <Wire.h>
10 #include <Adafruit_GFX.h>
11 #include <Adafruit_SSD1306.h>
12
13 // OLED setup
14 #define SCREEN_WIDTH 128
15 #define SCREEN_HEIGHT 64
16 #define OLED_RESET -1
17 #define SCREEN_ADDRESS 0x3C
18 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
19
20 // DHT setup
21 DHTesp dht;
22 float temperature, humidity;
23 unsigned long timeout = 0;
24
25 // Blynk
26 BlynkTimer timer;
27 WidgetLED LEDCONNECT(W0);
28 #define WIDGET_V1
29 #define DOOR_V2
30
31 // Buzzer setup (DS 4512)
32 #define BUZZER_PIN 14 // DS - GP1014
33 bool buzzerState = false;
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
```

```
#include <Adafruit_SSD1306.h>

// OLED setup

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64

#define OLED_RESET    -1

#define SCREEN_ADDRESS 0x3C

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

// DHT setup

DHTesp dht;

float temperature, humidity;

unsigned long timeShowOled = millis();

// Blynk

BlynkTimer timer;

WidgetLED LEDCONNECT(V0);

#define NHIETDO V1

#define DOAM V2

// Buzzer setup (Đã đổi sang D5)

#define BUZZER_PIN 14 // D5 - GPIO14

bool buzzerState = false;

// Fan (quạt) setup

#define FAN_PIN 12 // D6 - GPIO12
```

```
bool fanState = false;

// Forward declarations

void updateBlynk();

void showOled(float t, float h);

void controlOutputs();

// Setup

void setup() {

    Serial.begin(115200);

    delay(100);

    pinMode(BUZZER_PIN, OUTPUT);

    digitalWrite(BUZZER_PIN, LOW);

    pinMode(FAN_PIN, OUTPUT);

    digitalWrite(FAN_PIN, LOW);

    BlynkEdgent.begin();

    if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {

        Serial.println(F("SSD1306 allocation failed"));

        for(;;);

    }

    display.display();

    delay(2000);

    dht.setup(16, DHTesp::DHT11); // D0 -> DHT11

    timer.setInterval(1000L, updateBlynk);
```



```
}

// Main loop

void loop() {

  BlynkEdgent.run();

  timer.run();

  if (millis() - timeShowOled > dht.getMinimumSamplingPeriod()) {

    float t = dht.getTemperature();

    float h = dht.getHumidity();

    if (dht.getStatusString() == "OK") {

      temperature = t;

      humidity = h;

      showOled(temperature, humidity);

    }

    timeShowOled = millis();

  }

  controlOutputs();

}

// Update Blynk

void updateBlynk() {

  if (LEDCONNECT.getValue()) LEDCONNECT.off();

  else LEDCONNECT.on();

  Blynk.virtualWrite(NHIETDO, temperature);
```

```
Blynk.virtualWrite(DOAM, humidity);

}

// Show OLED

void showOled(float t, float h) {

    display.clearDisplay();

    // Vẽ nhiệt độ & độ ẩm

    display.setTextSize(2);

    display.setTextColor(SSD1306_WHITE);

    display.setCursor(0, 0);

    display.print("T: ");

    display.print(t, 1);

    display.print((char)247);

    display.println("C");

    display.print("H: ");

    display.print(h, 0);

    display.println("%");

    // Chuyển sang text size nhỏ để hiển thị trạng thái

    display.setTextSize(1);

    display.setCursor(0, 40);

    display.print("Buzzer: ");

    display.println(buzzerState ? "ON" : "OFF");

    display.setCursor(0, 50);
```

```
display.print("Fan  : ");

display.println(fanState  ? "ON" : "OFF");

display.display();

}

// Blynk write handlers

BLYNK_WRITE(V3) { // Buzzer control

    int pinValue = param.asInt();

    buzzerState = (pinValue == 1);

}

BLYNK_WRITE(V4) { // Fan (quạt) control

    int pinValue = param.asInt();

    fanState = (pinValue == 1);

}

// Control buzzer and fan hardware

void controlOutputs() {

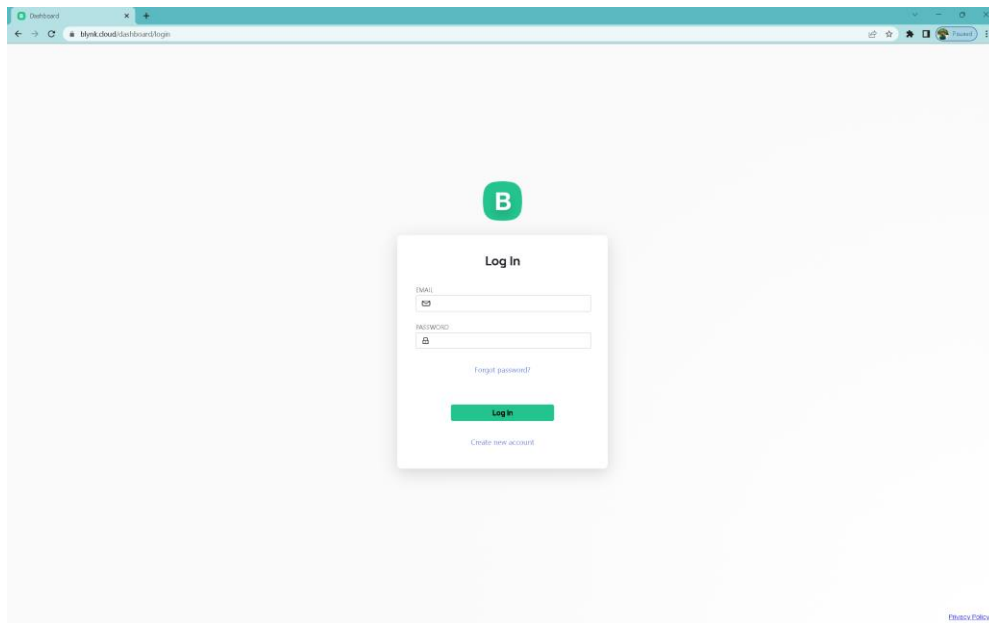
    digitalWrite(BUZZER_PIN, buzzerState ? HIGH : LOW);

    digitalWrite(FAN_PIN, fanState ? HIGH : LOW);

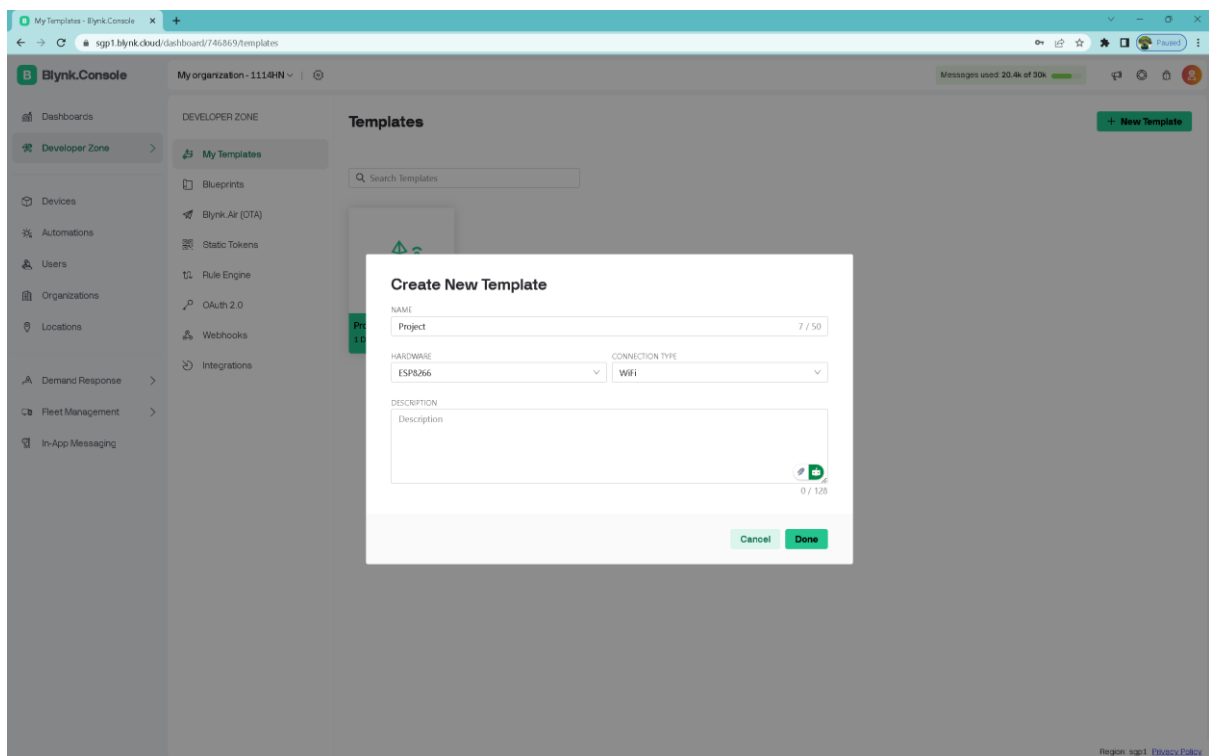
}
```

4.3. Các bước thực hiện:

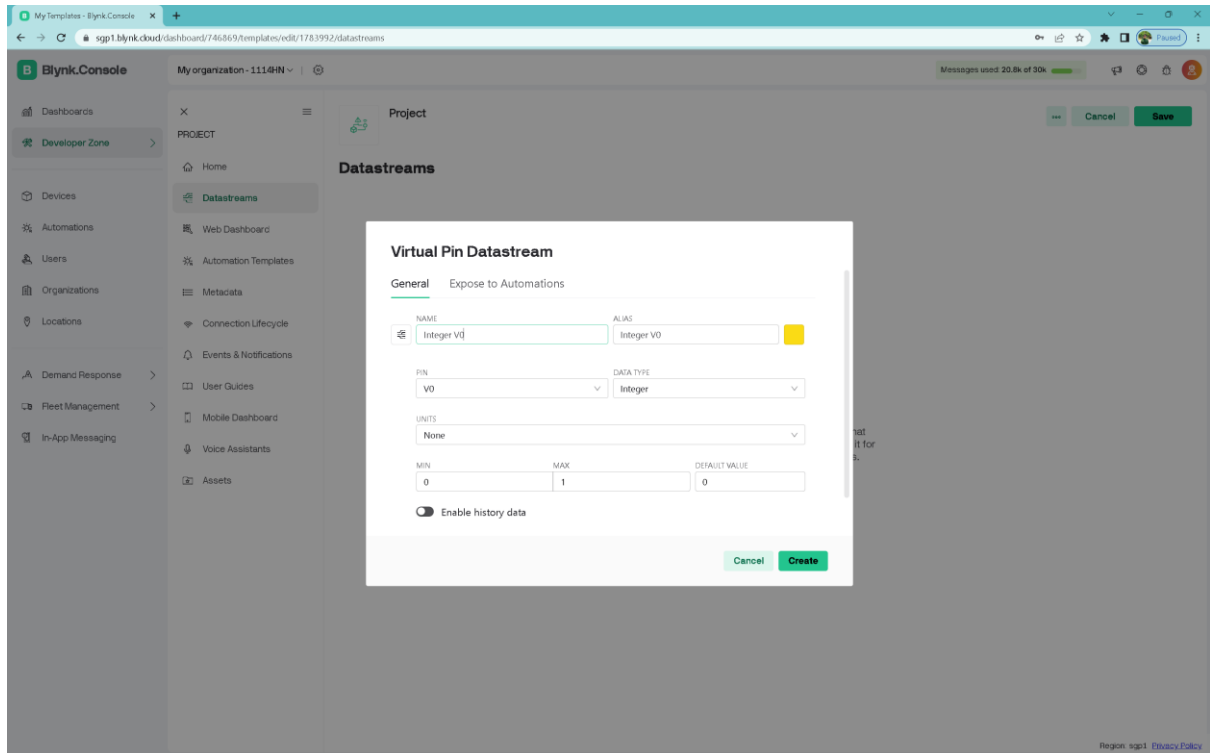
Bước 1: Tạo tài khoản trên web Blynk.cloud



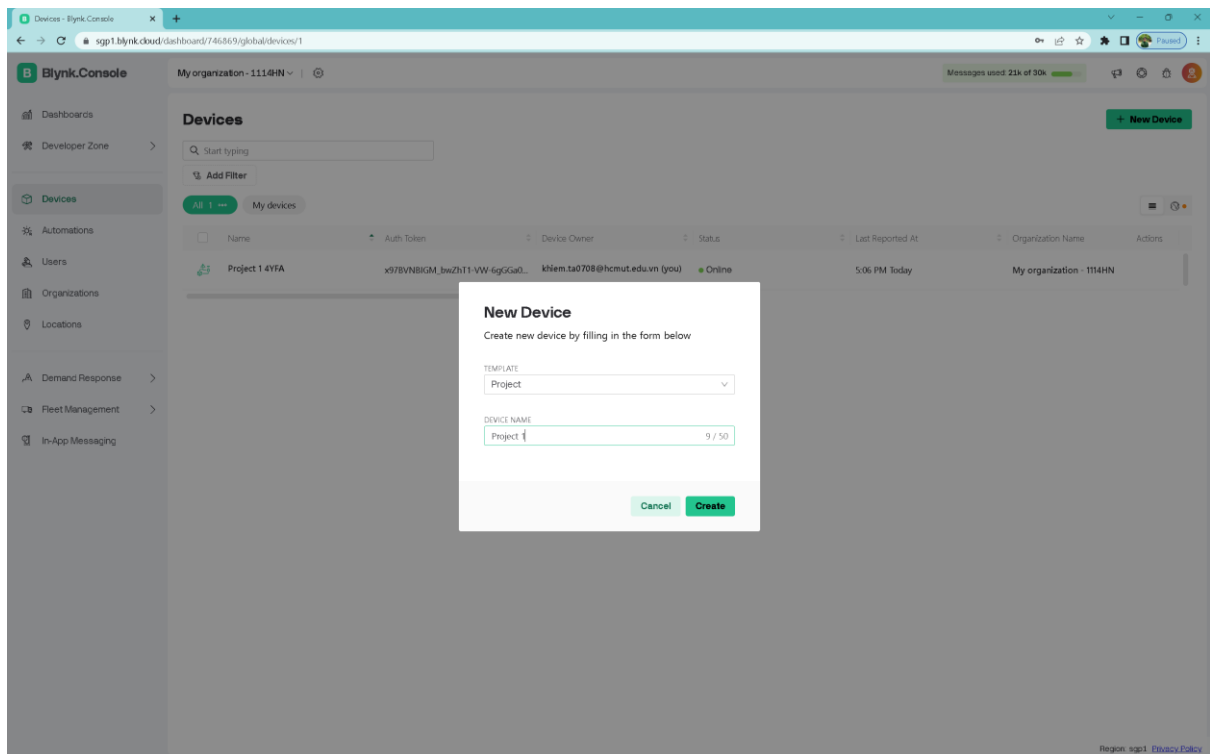
Bước 2: Tạo Template:



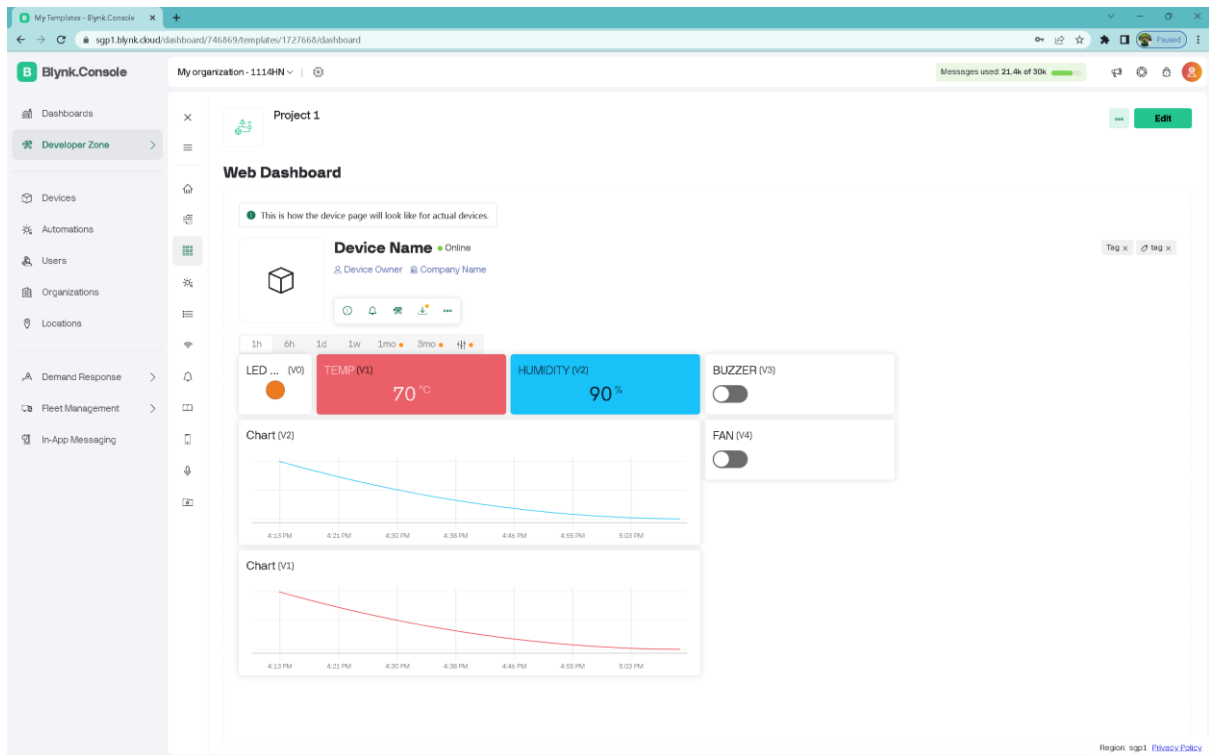
Bước 3: Thiết lập datastream:



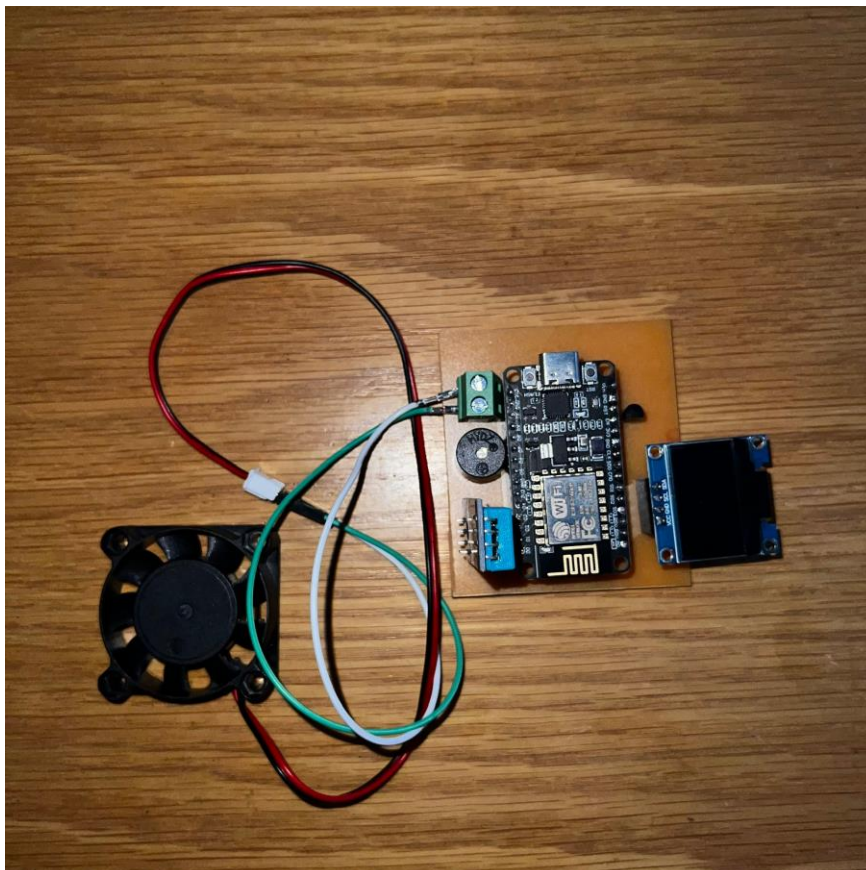
Bước 4: Thêm thiết bị mới từ Template



Bước 5: Thiết kế Web dashboard



Bước 6: Thực hiện đấu nối phần cứng theo Schematic.

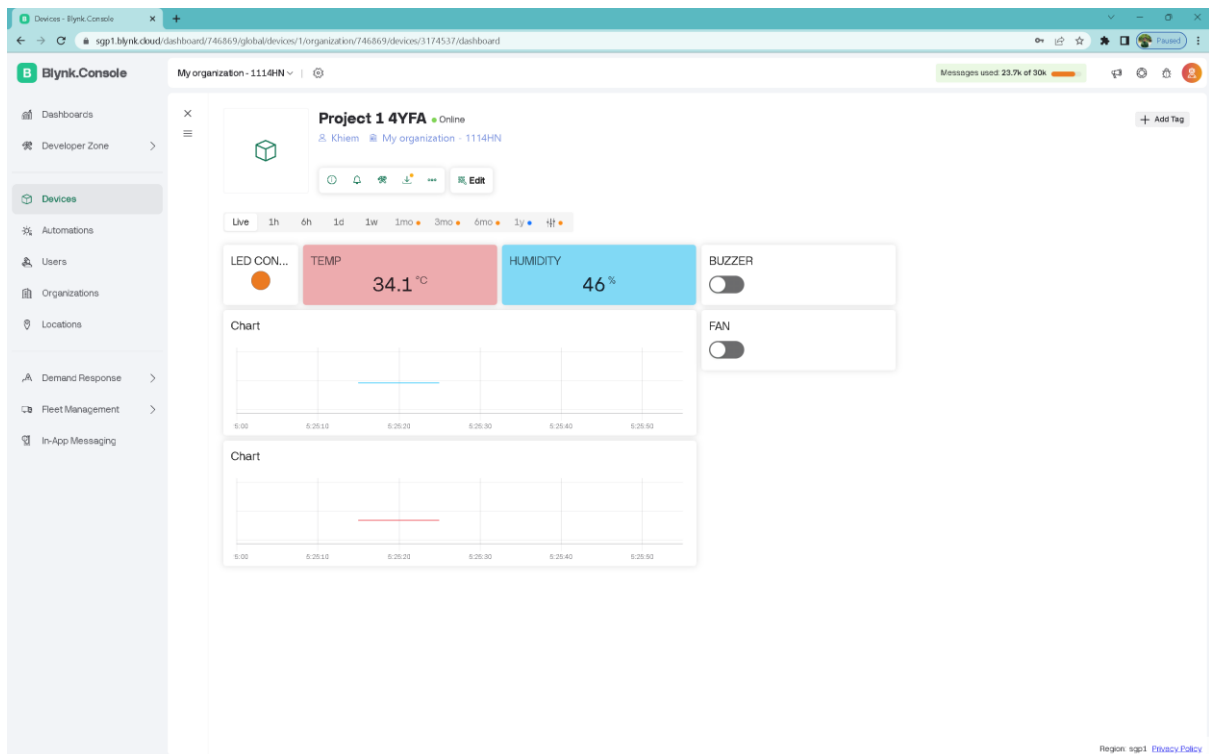


Bước 7: Tiến hành nạp code vào mạch.

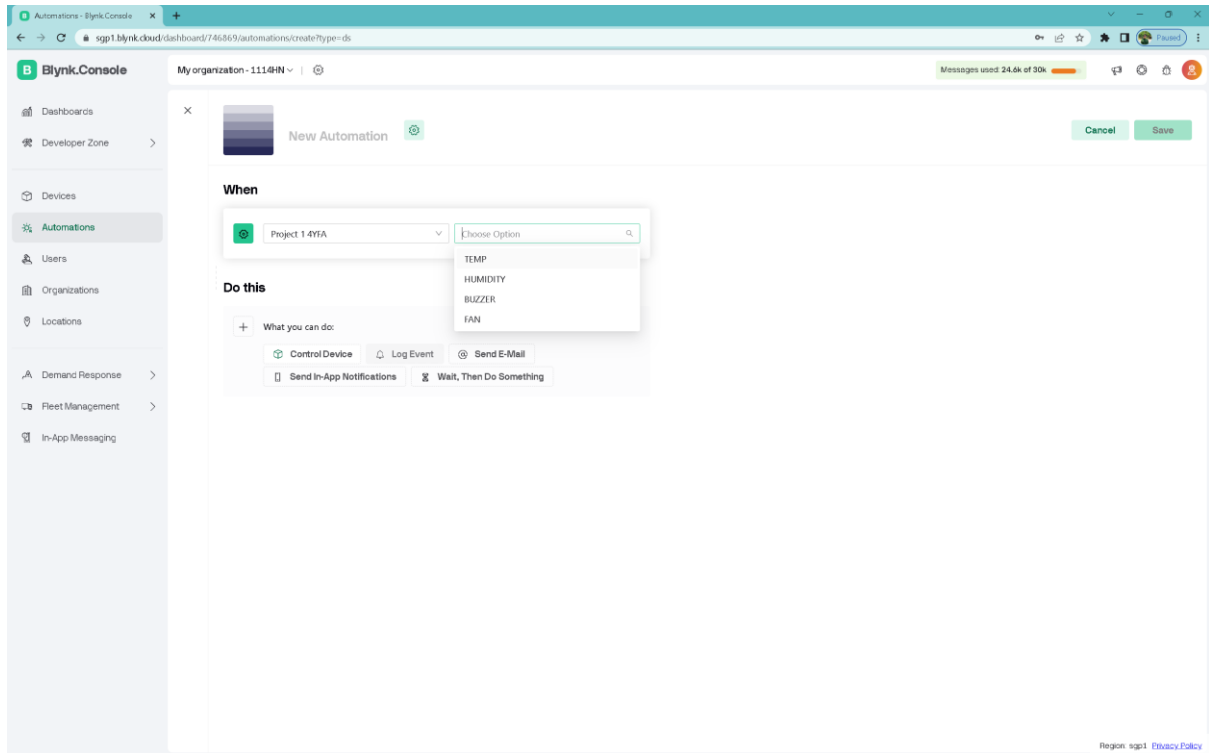
```

1 #define BLYNK_TEMPLATE_ID "T1PL6v917LqZ"
2 #define BLYNK_TEMPLATE_NAME "Project 1"
3 #define BLYNK_AUTH_KEY "a1a"
4 #define BLYNK_PRINT Serial
5 #define APP_DEBUG
6 #define USE_ARDUINO_BOARD
7 #include "BlynkEdgent.h"
8 #include "DHTesp.h"
9 #include <Wire.h>
10 #include <Adafruit_GFX.h>
11 #include <Adafruit_SSD1306.h>
12
13 // OLED setup
14 #define SCREEN_WIDTH 128
15 #define SCREEN_HEIGHT 64
16 #define OLED_RESET -1
17 #define SCREEN_ADDRESS 0x3C
18 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
19
20 // DHT setup
21 DHTesp dht;
22 float temperature, humidity;
23 unsigned long timeshowoled = millis();
24
25 // Blynk
26 BlynkTimer timer;
27 WidgetLED LEDCONNECT(v0);
28 #define NHITDO V1
29 #define DOAM V2
30
31 // Buzzer setup (bật dõ sang 0s)
32 #define BUZZER_PIN 14 // DS - GPIO14
33 bool buzzerState = false;
34
Output
Writing at 0x00020000... (40 %)
Writing at 0x00024000... (45 %)
Writing at 0x00028000... (50 %)
Writing at 0x0002c000... (54 %)
Writing at 0x00030000... (59 %)
Writing at 0x00034000... (63 %)
Writing at 0x00038000... (68 %)
Writing at 0x0003c000... (72 %)
Writing at 0x00040000... (77 %)
Writing at 0x00044000... (81 %)
Writing at 0x00048000... (86 %)
Writing at 0x0004c000... (90 %)
Writing at 0x00050000... (95 %)
Writing at 0x00054000... (100 %)
Wrote 49046 bytes (355402 compressed) at 0x00000000 in 31.3 seconds (effective 125.3 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
Done uploading
    
```

Bước 8: Kết nối với hệ thống qua Internet.



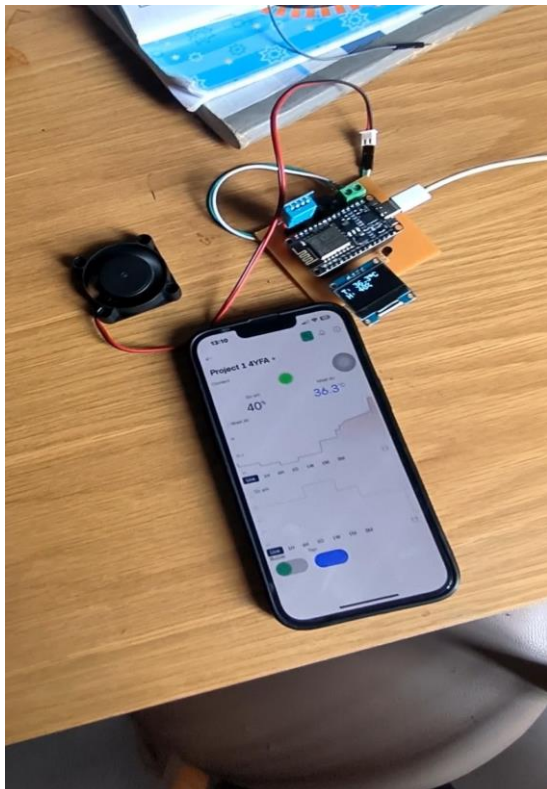
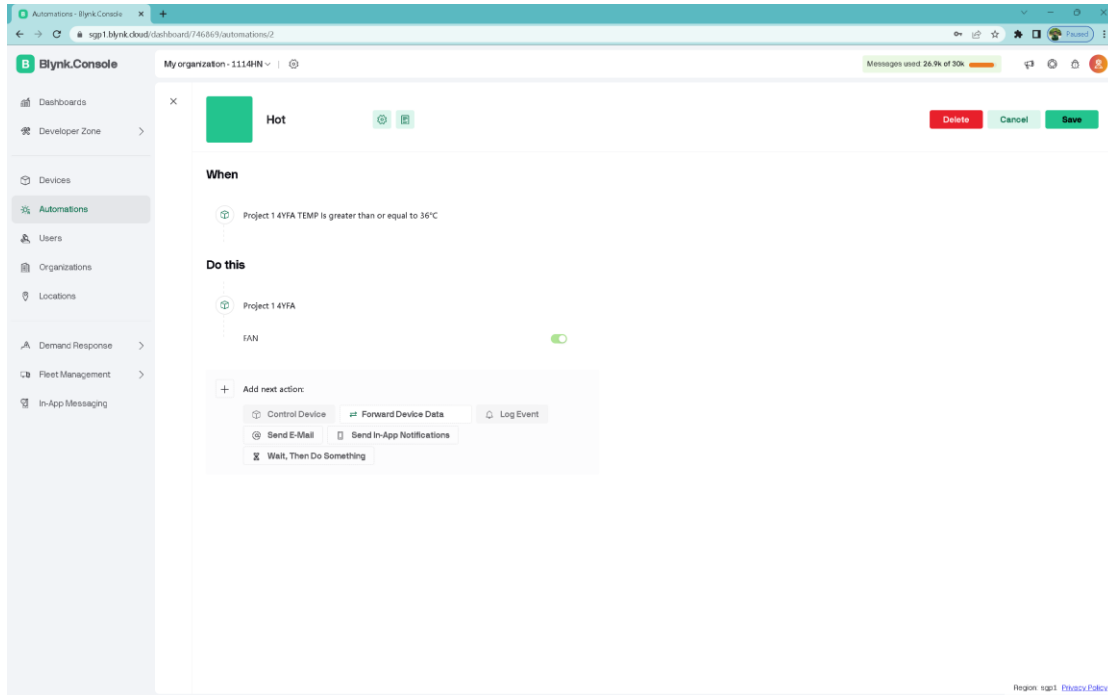
Bước 9: Thiết lập Automation để hệ thống thống hoạt động theo ý muốn.



4.4 Kết quả thực nghiệm:

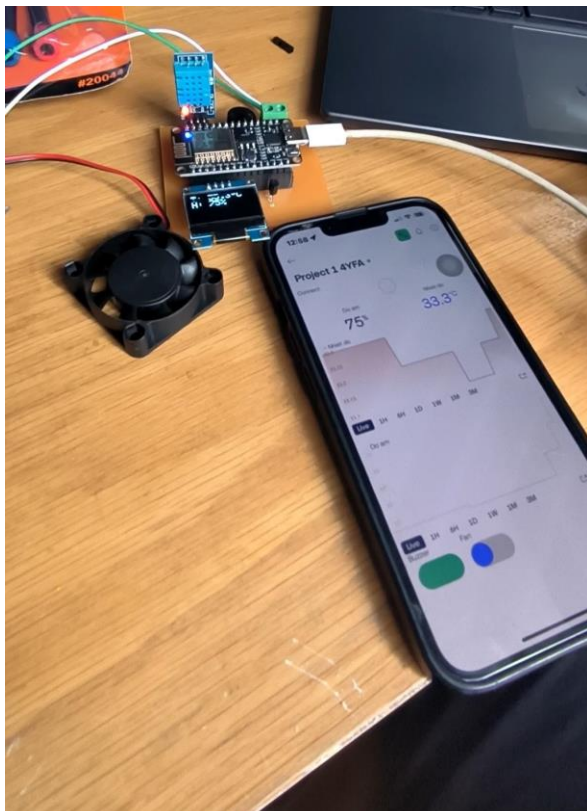
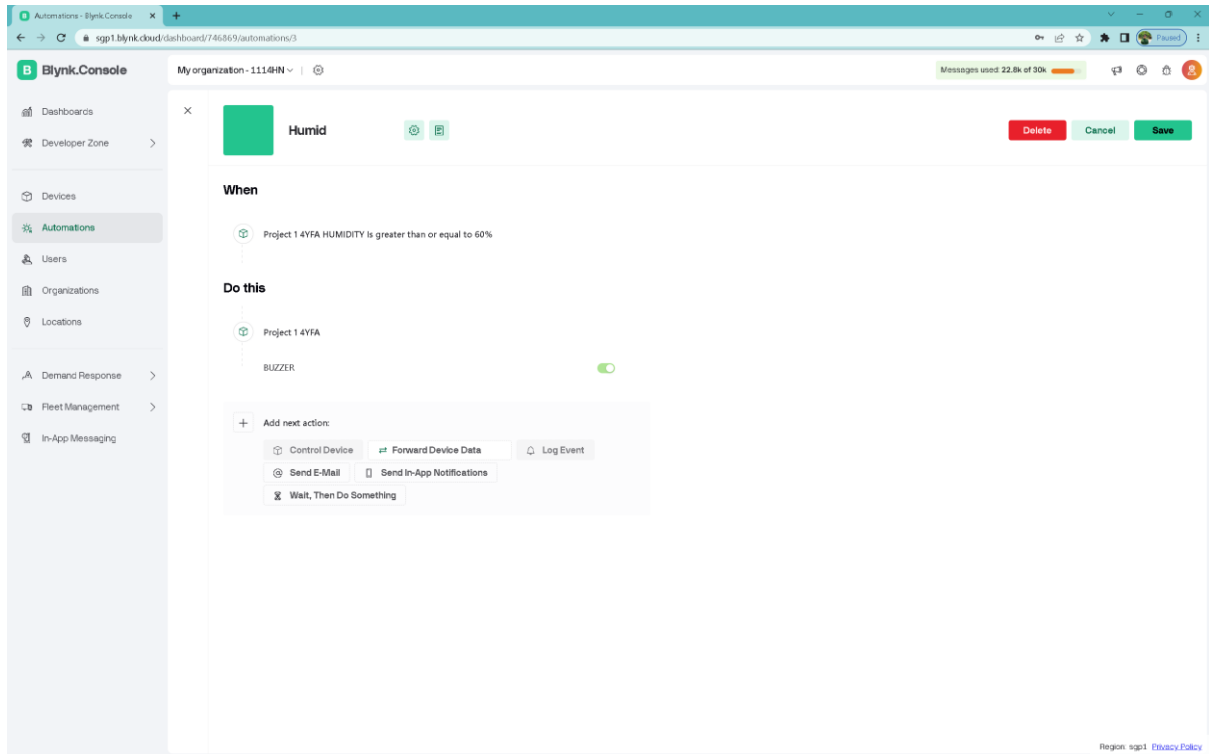
Trường hợp 1: Khi hệ thống vượt quá nhiệt độ:

Đặt giới hạn nhiệt độ là 36°C, khi hệ thống vượt quá hoặc đạt tới nhiệt độ này thì quạt sẽ quay:



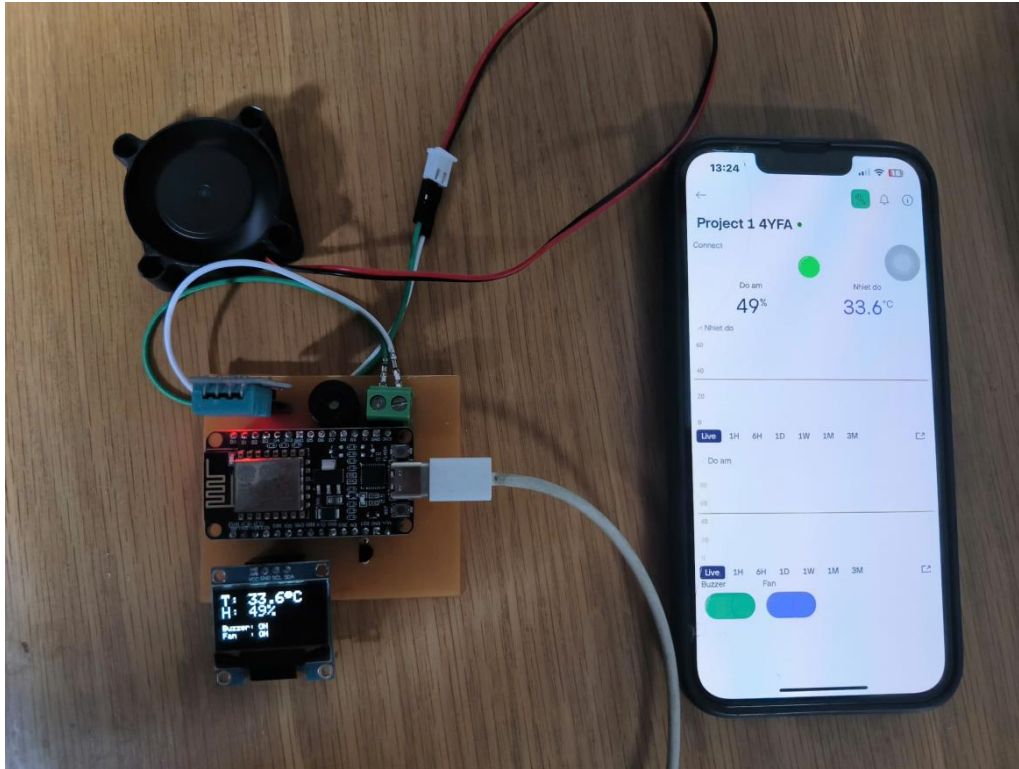
Trường hợp 2: Khi hệ thống vượt quá độ ẩm:

Đặt giới hạn độ ẩm là 60%, khi hệ thống vượt quá hoặc đạt tới độ ẩm này thì còi sẽ kêu:



Trường hợp 3: Khi hệ thống vượt quá nhiệt độ và độ ẩm:

Đặt giới hạn nhiệt độ là 32°C và độ ẩm là 48%, khi hệ thống vượt quá hoặc đạt tới nhiệt độ này thì quạt sẽ quay và còi sẽ kêu:

**5. Kết luận:****5.1 Kết luận**

Qua quá trình thực hiện đề tài "Hệ thống giám sát nhiệt độ qua Internet", sinh viên đã đạt được những kết quả và kinh nghiệm đáng kể. Hệ thống được xây dựng dựa trên vi điều khiển ESP8266, cảm biến DHT11, tích hợp nền tảng Blynk IoT và các thiết bị ngoại vi như quạt, buzzer đã hoạt động ổn định, đáp ứng được mục tiêu đề ra:

- Giám sát từ xa : Dữ liệu nhiệt độ, độ ẩm được hiển thị trên ứng dụng Blynk và màn hình OLED theo thời gian thực.
- Tự động cảnh báo : Hệ thống kích hoạt quạt và buzzer khi nhiệt độ/độ ẩm vượt ngưỡng, đảm bảo phản ứng kịp thời.
- Kết nối Internet : ESP8266 duy trì kết nối WiFi ổn định, cho phép điều khiển và giám sát qua Internet.

Kinh nghiệm thu được :

Hiểu sâu về nguyên lý hoạt động của cảm biến DHT11, ESP8266 và cách tích hợp chúng vào hệ thống IoT.

Rèn luyện kỹ năng lập trình (Arduino IDE), thiết kế mạch điện, và xử lý sự cố phần cứng.

Nắm vững quy trình triển khai ứng dụng IoT thông qua nền tảng Blynk.

Ưu điểm :

- Chi phí thấp, dễ triển khai nhờ sử dụng linh kiện phổ biến.
- Giao diện người dùng trực quan, dễ điều khiển từ xa.

Hệ thống linh hoạt, có thể mở rộng bằng cách thêm cảm biến hoặc thiết bị khác.

Khuyết điểm :

- Độ chính xác của DHT11 còn hạn chế ($\pm 2^{\circ}\text{C}$ cho nhiệt độ, $\pm 5\%$ RH cho độ ẩm).
- Phụ thuộc vào kết nối Internet và máy chủ Blynk, gây gián đoạn nếu mất mạng.
- Phạm vi kết nối WiFi chỉ đạt khoảng 30m trong điều kiện lý tưởng.

So sánh với mục tiêu ban đầu :

Đề tài đã đáp ứng đầy đủ các yêu cầu về giám sát, cảnh báo và điều khiển từ xa. Tuy nhiên, độ chính xác của cảm biến chưa cao so với thiết bị thương mại, cần cải thiện trong tương lai.

5.2 Hướng phát triển

Đề tài mở ra nhiều hướng phát triển và ứng dụng tiềm năng:

Nâng cấp phần cứng :

Thay thế DHT11 bằng cảm biến chính xác hơn (DHT22, BME280) để cải thiện độ tin cậy.

Tích hợp thêm cảm biến khí CO₂, ánh sáng hoặc cảm biến chuyển động để mở rộng chức năng giám sát.

Sử dụng vi điều khiển ESP32 để tăng khả năng xử lý và kết nối Bluetooth.

Cải tiến phần mềm :

Xây dựng ứng dụng web riêng thay vì phụ thuộc vào Blynk, giúp chủ động hơn trong quản lý.

Áp dụng thuật toán AI để dự đoán xu hướng nhiệt độ và điều khiển thiết bị thông minh.

Triển khai giao thức MQTT để tối ưu hóa truyền dữ liệu và tiết kiệm năng lượng.

5.3 Ứng dụng thực tiễn :

Nhà thông minh: Tự động điều chỉnh nhiệt độ phòng, cảnh báo cháy nổ.

Nông nghiệp: Giám sát điều kiện nhà kính, tự động tưới tiêu và thông gió.

Công nghiệp: Theo dõi nhiệt độ máy móc, phòng ngừa quá tải.

Với tính linh hoạt và khả năng mở rộng, hệ thống không chỉ dừng lại ở việc giám sát nhiệt độ mà còn có thể trở thành nền tảng cho các giải pháp IoT đa dạng trong tương lai.

6. Tài liệu tham khảo:

1. Blynk Inc., "Blynk IoT Platform Documentation," 2023. <https://docs.blynk.io/>
2. Espressif Systems, "ESP8266 Technical Reference Manual."
https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf.
3. Adafruit Industries, "DHT11 Basic Temperature-Humidity Sensor Datasheet."
<https://cdn-shop.adafruit.com/datasheets/DHT11-chinese.pdf>.
4. Arduino, "Arduino IDE Installation Guide," 2024. Online].
<https://www.arduino.cc/en/software>.
5. T. V. On, "Lập trình IoT với ESP8266 và Arduino," Nhà xuất bản Khoa học Kỹ thuật, Hà Nội, 2023.
6. F. Fentanes, "IoT Projects with ESP8266: Build IoT Applications Using Sensors, Cloud Services, and APIs," Packt Publishing, 2022.
7. M. Schwartz, "Internet of Things with ESP8266," CreateSpace Independent Publishing Platform, 2021.
8. Nordic Semiconductor, "nRF24L01+ Single Chip 2.4GHz Transceiver Product Specification,"
https://infocenter.nordicsemi.com/pdf/nRF24L01P_PS_v1.0.pdf.
9. Texas Instruments, "LM7805 Voltage Regulator Datasheet," 2023.
<https://www.ti.com/lit/ds/symlink/lm7805.pdf>.